Question 3:

(a)(i)

Firstly, we found the probability of the classes π_c:

```
{'CYT': 0.313,
 'ERL': 0.005,
 'EXC': 0.023,
 'ME1': 0.03,
 'ME2': 0.038,
 'ME3': 0.106,
 'MIT': 0.159,
 'NUC': 0.292,
 'POX': 0.015,
 'VAC': 0.019}
```

Then, we found the mean percent classification error on the training set, on the test set, the standard deviation of the percent classification error on the training set and on the test set:

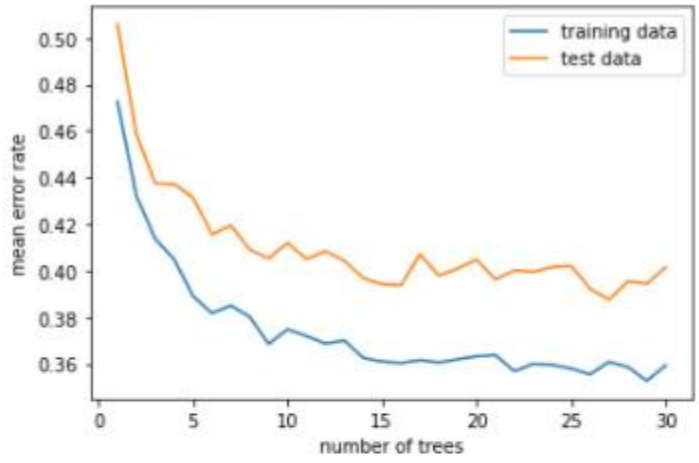
```
mean classification error on training data 0.7758
standard deviation of classification error on training data 0.015477725931156698
----------
mean classification error on test data 0.7774793388429753
standard deviation of classification error on test data 0.019448943139650756
```

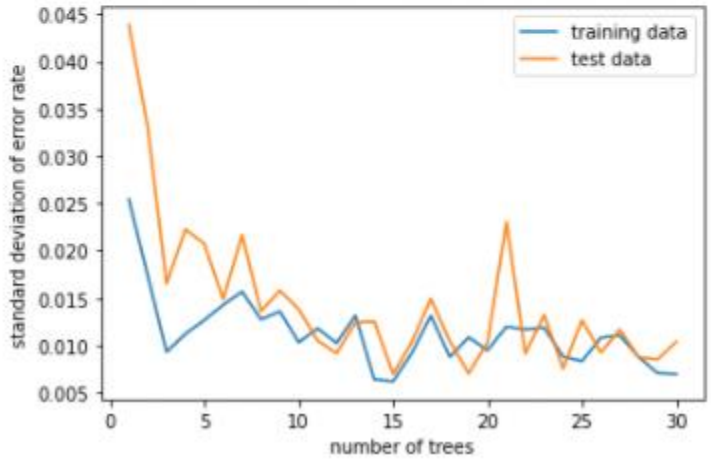
(a)(ii)

The most populated class is CYT with probability of 0.313 as shown above, and if were to find the percent classification error on the training set and separately on the test set for a classifier that only outputs this label, we will get:

```
classification error on training data 0.687
classification error on test data 0.6900826446280992
```

(b)(i)

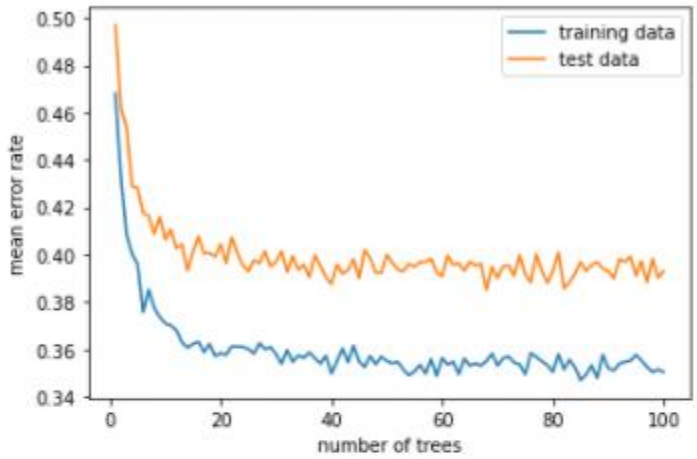


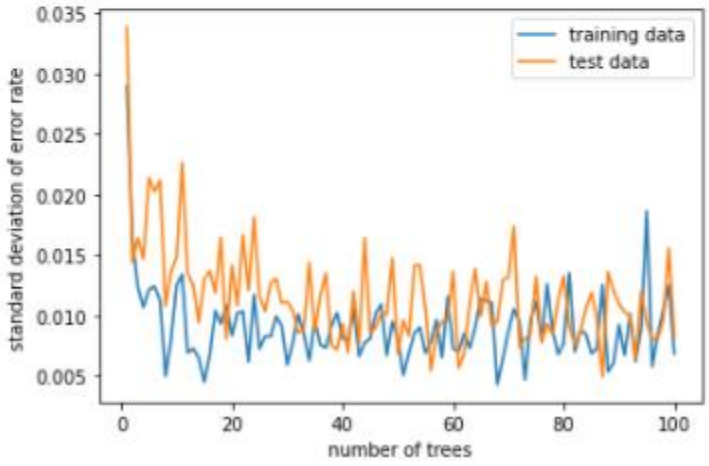


The lowest test error is when B= 27 with test error of: 0.387603305785124 with standard deviation of 0.011658459487327175

as observed in the plot, both training and test mean error rates reduce as the number of trees increases. The rate of the reduction in the error rates decreases, and approximately saturates when the number of trees gets big (about 18). For the standard deviation plot, it seems that there is no obvious tendency to decrease or increase after number of trees exceeds approximately=8. Also, it might be inferred from the plot that the training error keeps reducing more smoothly with the number of trees, compared with the test error.

(ii)



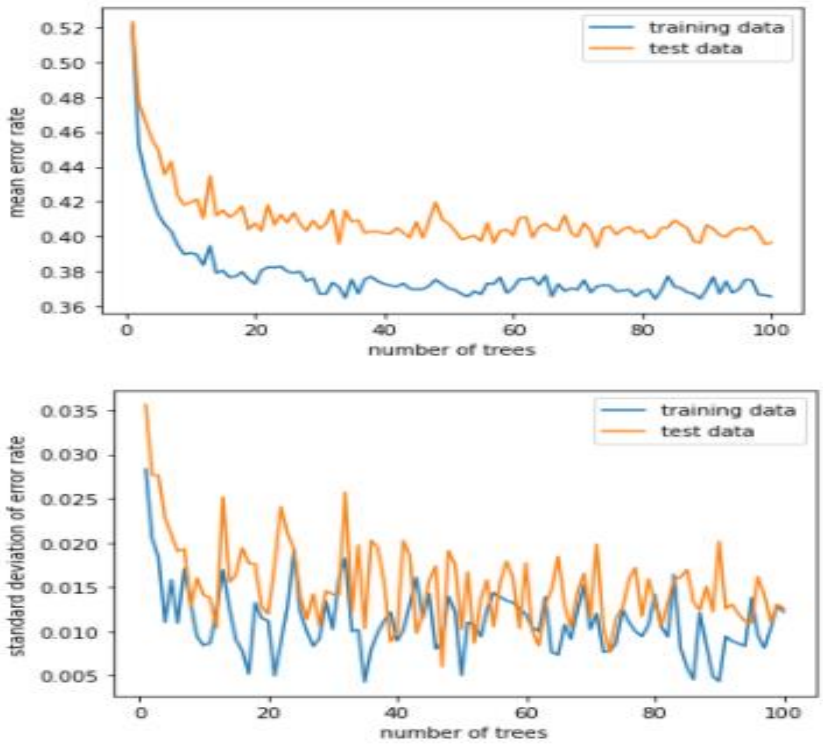


the lowest test error is when B= 68 with test error of: 0.3851239669421488 with standard deviation of 0.009522081505622249

It seems that there are no considerable changes in the errors after the number of trees exceeds a certain number, say roughly 40. For number of trees approximately >40 both test and train errors fluctuate within relatively small range. So, yes it shows convergence. For the standard deviation, similar to the previous case, there is no upward or downward patterns after exceeding certain number of trees.

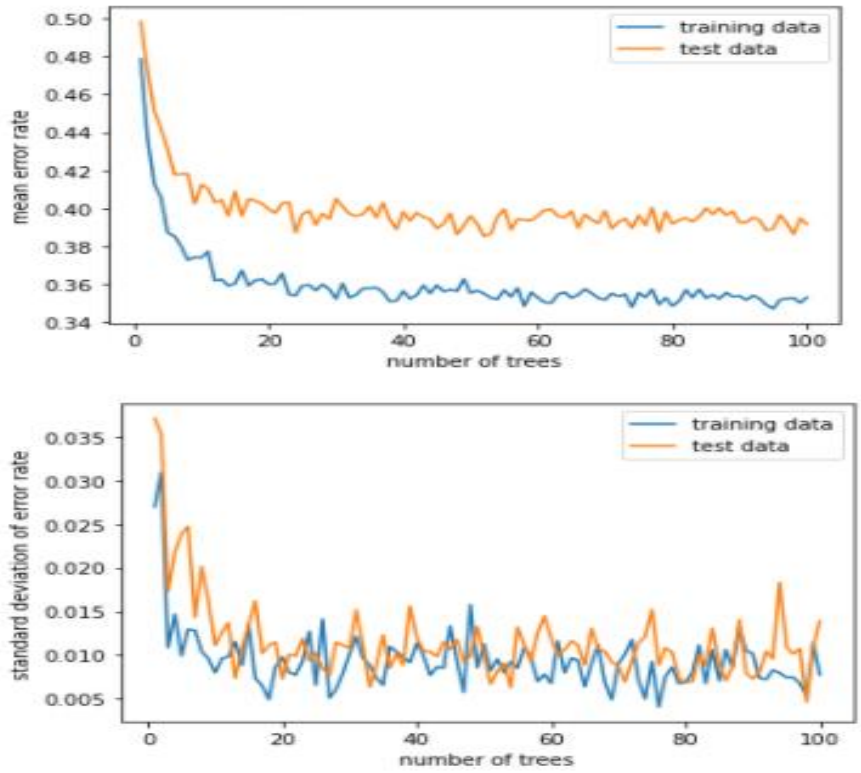
(iii)

Setting 1: Number of trees=100, bag size=1/3, max depth=5, max features=3



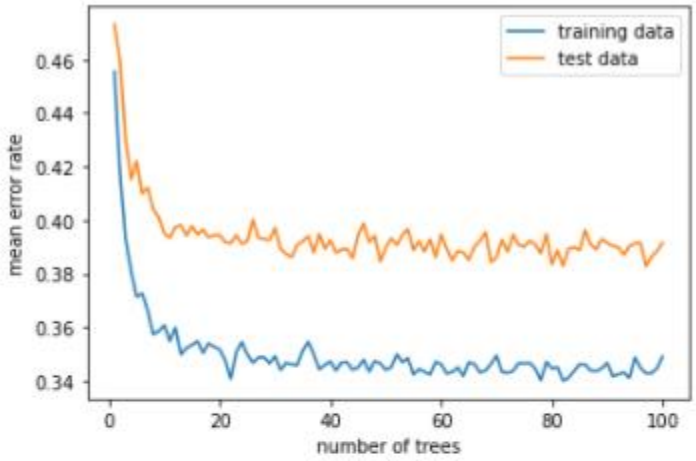
the lowest test error is when B= 73 with test error of: 0.393801652892562 with standard deviation of 0.007686394726338133

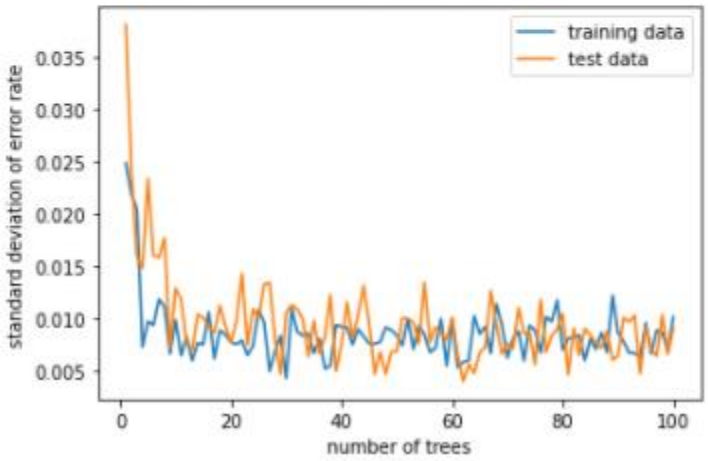
Setting 2: Number of trees=100, bag size=1/2, max depth=5, max features=3



the lowest test error is when B= 52 with test error of: 0.3853305785123967 with standard deviation of 0.008121175209517979

Setting 3: Number of trees=100, bag size=2/3, max depth=5, max features=3

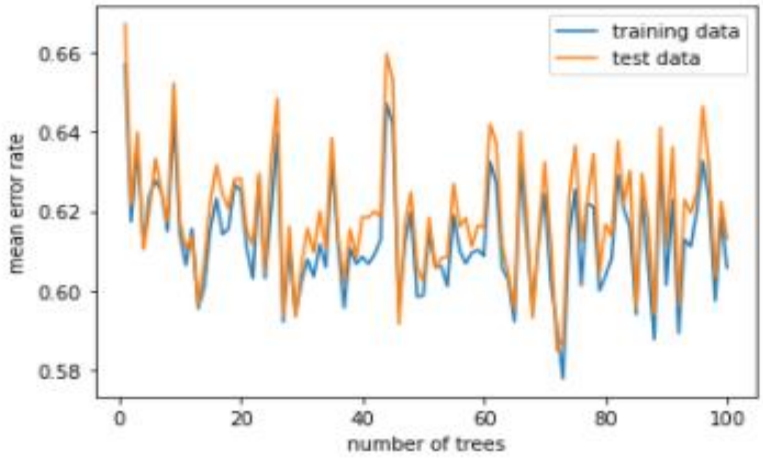


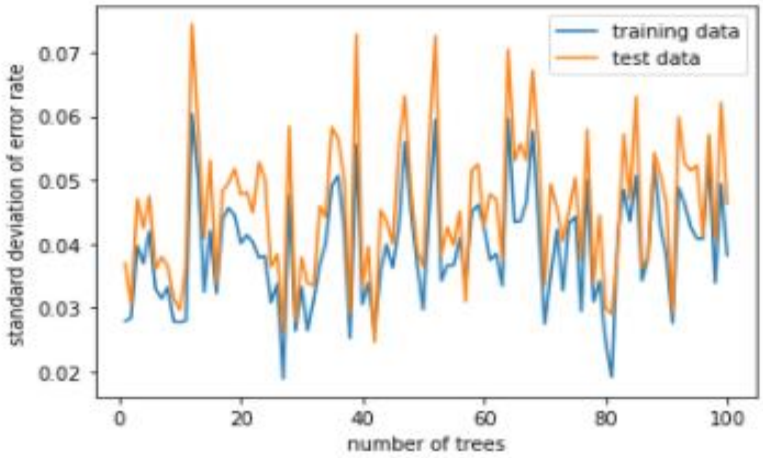


the lowest test error is when B= 97 with test error of: 0.3828512396694215 with standard deviation of 0.006637959359577682

as seen in the results above, the minimum test error decreases as we increase the bag size, which might be intuitive that the model will generalize better if we train it with more data. Also, it seems that the standard deviation curve had less fluctuation as the bag size increases.

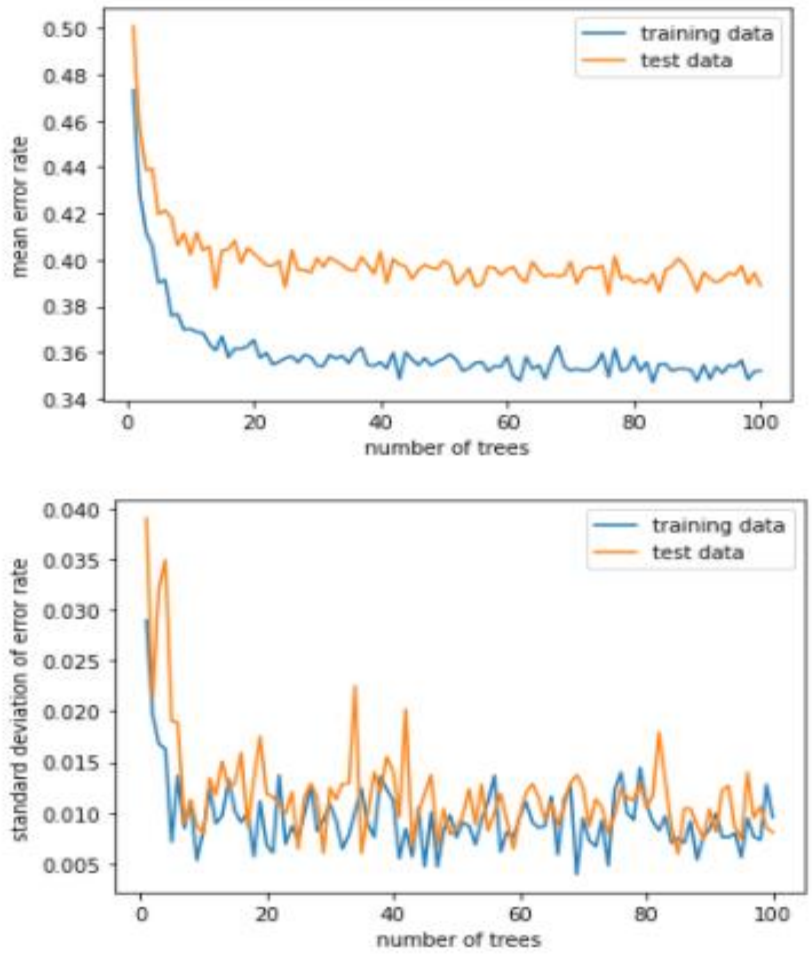
Setting 4: Number of trees=100, bag size=1/2, max depth=1, max features=3





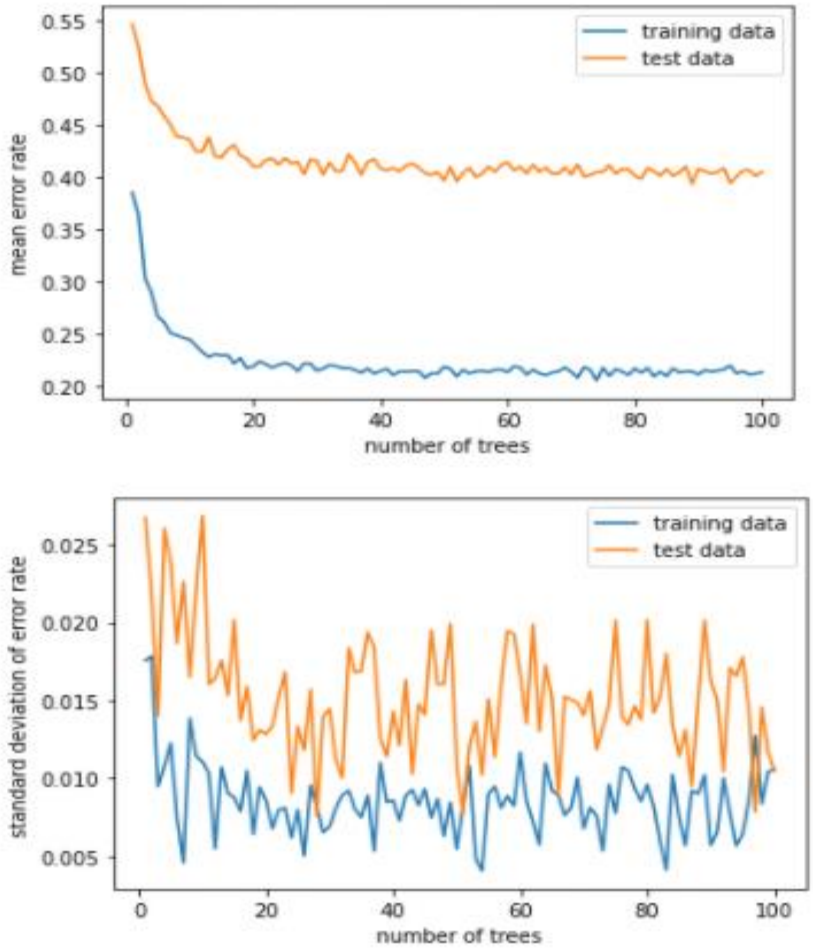
the lowest test error is when B= 72 with test error of: 0.5849173553719009 with standard deviation of 0.04549631814581051

Setting 5: Number of trees=100, bag size=1/2, max depth=5, max features=3



the lowest test error is when B= 76 with test error of: 0.3851239669421488 with standard deviation of 0.012334555005453964

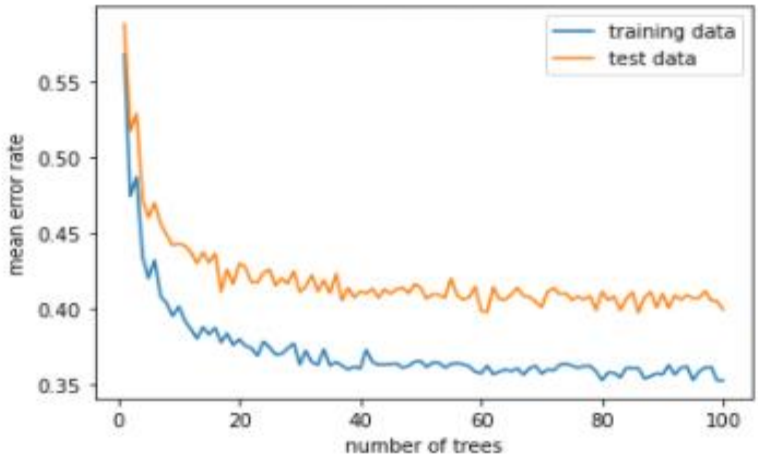
Setting 6: Number of trees=100, bag size=1/2, max depth=None, max features=3

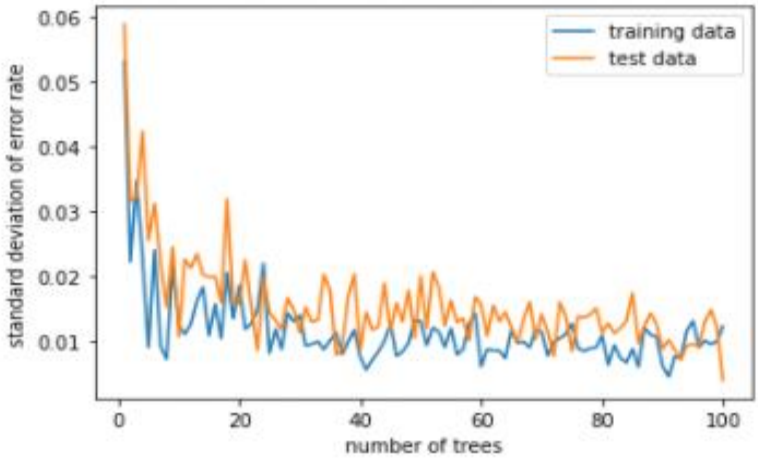


the lowest test error is when B= 89 with test error of: 0.3933884297520661 with standard deviation of 0.02014224397843628

from the results above, it seems that using a one node trees causes both training and test error to be high, and they don't decrease with the number of trees. Similarly for the standard deviation, it has high fluctuation and doesn't tend to increase nor decrease with the number of trees. When we increase the trees' depth, however, both errors significantly decrease and they also show a decreasing trend with the number of trees, and the standard deviation decreases and shows a lower fluctuation. When we have unlimited depth, the training error heavily reduced (which is intuitive as the model has more "freedom" to fit the data. The test error, however, experienced a little increase, which might be because the model overfitted the training data and hence had a slightly worse generalization. The standard error of the training also reduces, but the standard deviation of the test error rate became slightly wigglier. We can also notice that the mean errors have lower fluctuations compared to the previous 2 cases.

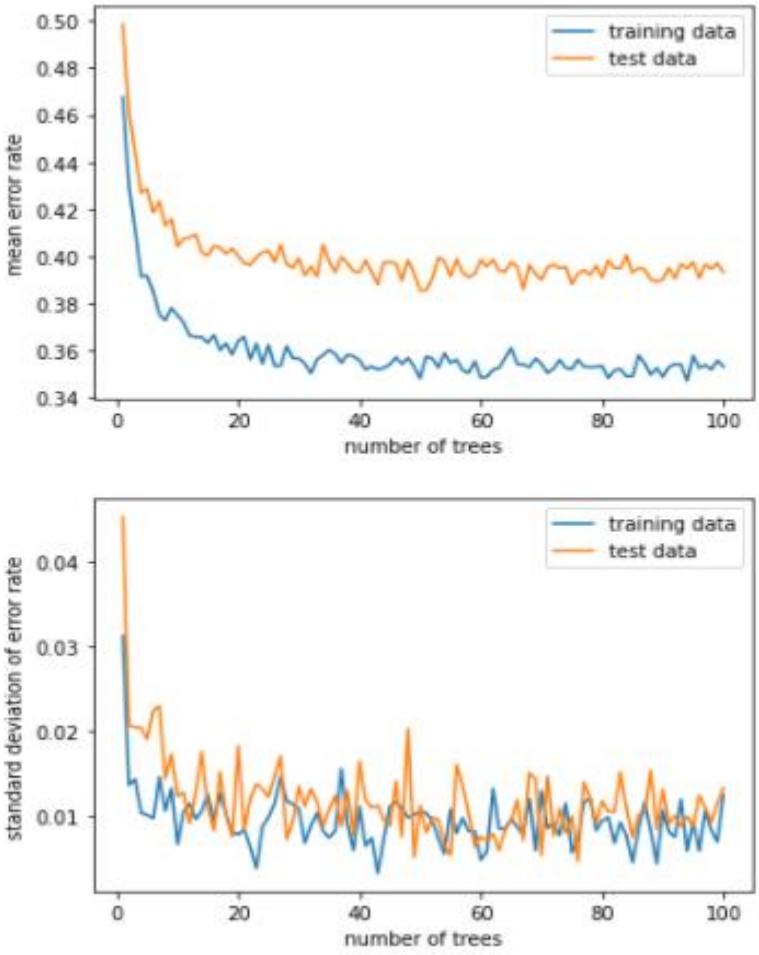
Setting 7: Number of trees=100, bag size=1/2, max depth=5, max features=1





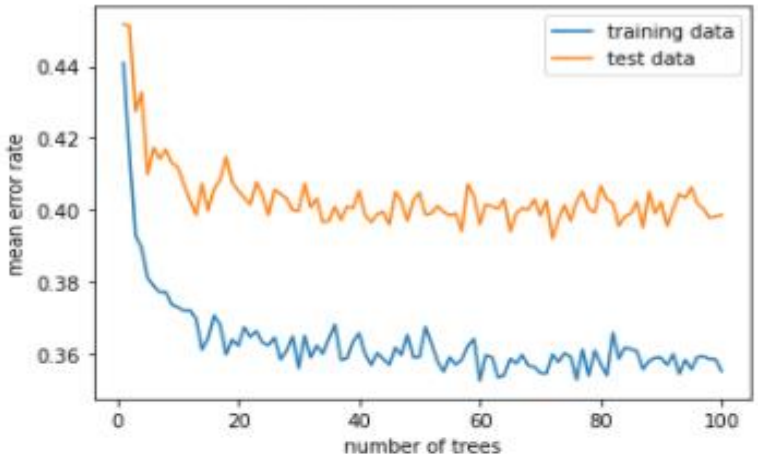
the lowest test error is when B= 86 with test error of: 0.39731404958677685 with standard deviation of 0.00978082675912778

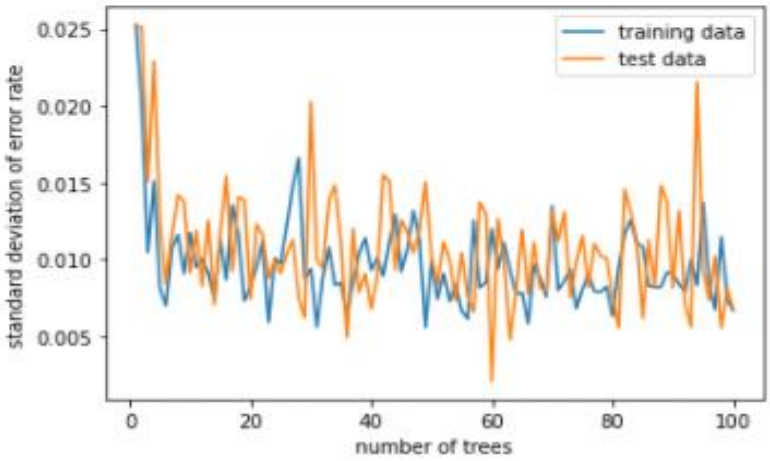
Setting 8: Number of trees=100, bag size=1/2, max depth=5, max features=3



the lowest test error is when B= 50 with test error of: 0.3853305785123967 with standard deviation of 0.011174229159495827

Setting 9: Number of trees=100, bag size=1/2, max depth=5, max features=8





the lowest test error is when B= 72 with test error of: 0.39214876033057855 with standard deviation of 0.013093371502276768

It seems that there is no difference in the test errors, since they are in range of the corresponding standard deviation. However, there is a slight decrease in the test error when increased the max feature to split from 1 to 3.

(iv)

Lots of models approximately performed equally well in terms of generalization performance when we take standard deviation into account when we compare test errors. Since there are no considerable changes in a lot of parameters changes, we can say that the classifier is robust.

Setting 3 slightly performed best, since it has low mean error rate and standard deviation.

Setting 3 has mean test error rate of 0.3828512396694215

Trivial classifier 1 has test error rate of 0.7774793388429753

Trivial classifier 2 has test error rate of 0.6900826446280992

Comparing those results, clearly the model has learned from the data and had a better generalization performance